Frame Synchronization in Time-Multiplexed PCM Telemetry With Variable Frame Length

U. Timor
Communications Systems Research Section

New methods of parallel and serial synchronization for time-multiplexed phase-coherent telemetry signals are presented. In the parallel case, i.e., when the sync code is transmitted on a separate channel, M synchronization codes are generated by concatenating a common pseudonoise (PN)-like sequence of short length with words from a self-synchronizing code. The frame synchronizer can lock on any of the M codes, which is particularly useful when frames of different lengths have to be transmitted. In the serial case, that is, when each frame starts with an identical sync code, false sync due to replicas of the code randomly generated by the data is completely eliminated by transferring the sync code to the quadrature channel, while the data are transmitted on the in-phase channel.

I. Introduction

In time-multiplexed pulse-code-modulated (PCM) telemetry, binary data signals from several sources are grouped into frames, which have to be identified at the receiver in order to demultiplex the data. Frame synchronization is obtained by inserting in series (e.g., at the beginning of each frame), or in parallel (i.e., on a separate sync channel), a frame sync code (FSC). At the receiver, the frame synchronizer correlates the received signal with its own replica of the FSC, for different bit shifts, until synchronization is acquired. The synchronization is then maintained by verifying the repetition of this code at each frame, provided the frame length is fixed.

In some future cases, such as would occur in the adaptive telemetry era in the late 1970s and beyond, data rates and the number of data sources are not fixed, and it is advantageous to change the frame length and data format. In this case, at any instant of time, one of M possible frames is transmitted, and the receiver has to detect which frame is being sent while acquiring synchronization. However, standard methods of serial and parallel synchronization are inefficient for this task. In parallel synchronization, the FSC is usually a PN sequence (or a combination of several PN sequences) of total length equal to the frame length, and thus frames of different lengths have different PN sequences associated with them. The frame synchronizer correlates the received FSC with the various synchronization codes to determine which one is being transmitted, resulting in an increased complexity and degraded performance. In serial synchronization, the main problem is that of false sync due to replicas or "almost replicas" of the FSC, generated by the random data. The probability of false sync can be reduced by verifying the occurrence of the FSC on successive frames, but only if the frame length is known.

We will present new methods for parallel and serial synchronization, which avoid the described problems, and which can be used in multiframe systems. In the parallel synchronization case, a common PN-like sequence of short length (typically, two orders of magnitude smaller than the frame length) is used for all frames, and the FSC is generated by concatenating this sequence with words from a self-synchronizable code. In the serial case, the false sync due to random data is completely eliminated by transmitting the FSC, serially, on the quadrature channel, while the data are sent on the in-phase channel. For both methods, the acquisition time is considerably reduced.

II. Parallel Synchronization

In parallel synchronization, the time-multiplexed data and the FSC are frequency multiplexed, that is, both data and sync are transmitted simultaneously on different subcarriers. Since the power allocated to the sync is always small compared to the data power, the loss due to crossmodulation between the sync and the data is negligible. For example, if the power allocated to the sync channel is 5% of the data power, and 5% of the total power is transmitted as RF reference, the crossmodulation loss, using interplex (Ref. 1), is only 0.25%.

If the frame length N is fixed, the synchronization code can be a PN sequence of length N, which has an in-phase correlation of N and all out-of-phase periodic correlations of -1. The frame synchronizer correlates the incoming code with its own replica, for various bit shifts, and acquires synchronization using maximum-likelihood or threshold decision rules. If there are M possible frames, each with its own PN sequence, the frame synchronizer has to correlate the incoming code with all PN sequences to find the one which can be synchronized. This slows the acquisition and increases the complexity of the receiver.

In the next section we will describe a two-step multiframe synchronization method, where at the first step partial synchronization is obtained, no matter which frame is sent, while at the second step a self-synchronized decoder determines the frame and completes the synchronization. The only constraint on the frame lengths is that all of them are divisible by a common integer K, typically 2 orders of magnitude smaller than the frame lengths.

III. Parallel Synchronization for Several Frame Lengths

Let $F_1, \cdots, F_M$ be M different frame formats having lengths N_i, respectively, where

$$N_i = m_i K, \qquad i = 1, \cdots, M \tag{1}$$

and min $(m_i) >> 1$.

Typically, K is of the order of 15–25 and $N_i > 1000$, so that condition (1) is not a severe restriction on the N_i's.

Let $D = \{d_i\}$ be a self-synchronizable binary code dictionary of size M and word lengths $W_i = m_i$, $i = 1, \cdots, M$, i.e., $d_i = \{d_{i1}, \cdots, d_{im_i}\}$ and $d_{ij} = \pm 1$.

Finally, let C be a binary sequence of length K, to be chosen later. The frame synchronization code S_i for F_i is the concatenation of d_i and C, that is,

$$S_i = d_{i1}C, d_{i2}C, \cdots, d_{im_i}C \tag{2}$$

Clearly, S_i is a binary sequence of length $N_i = Km_i$, composed of m_i successive $\pm C$.

The acquisition is performed in two steps:

(1) Step I: C-synchronization

Let C be chosen such that the absolute values of all $2(K-1)$ out-of-phase correlations of C with (C, C) and $(C, -C)$ (Fig. 1) are small compared to K, which is the inphase autocorrelation of C.

PN sequences yield small out-of-phase correlations for the (C, C) case, but not necessarily for the $(C, -C)$ one. Let $r(\ell)$ be the partial correlation of C when shifted ℓ bits to the right (Fig. 2).

The out-of-phase correlations of C with (C, C) and $(C, -C)$ are, respectively,

$$\left.\begin{aligned} R_+(\ell) &= r(\ell) + r(-\ell) \\ R_-(\ell) &= r(\ell) - r(-\ell) \end{aligned}\right\} \quad \ell = 1, \cdots, K-1 \tag{3}$$

and their maximum absolute value is

$$R \triangleq \max_{1 \leq \ell \leq K-1} \{ |R_+(\ell)|, |R_-(\ell)| \} \tag{4}$$

Given K, we want to find sequences C which yield the smallest possible R. A necessary condition for a sequence

C to have a small R is that

$$\hat{R} \triangleq \max_{-(K-1)\leq \ell \leq K-1} |r(\ell)| \tag{5}$$

is small, so the search can be narrowed to sequences which have small $\hat{R}$.

Case 1: $\hat{R} = 1$

These are Barker sequences (Ref. 2), which exist for $K = 3, 5, 7, 11$, and 13. All of them yield $R = 1$.

Case 2: $\hat{R} = 2$

Such sequences have been found for all lengths $K \leq 21$ and $K = 25$ and 28 (Ref. 3) yielding the following R's:

$R = 2$, for $K = 14$ and 18

$R = 3$, for $K = 15, 17, 19, 21$, and 25

$R = 4$, for $K = 16, 20$, and 28

Case 3: $\hat{R} = 3$

Such sequences are known for all lengths $K \leq 34$ (Ref. 3) and yield the following R's:

$R = 4$, for $K = 24$ and 26

$R = 5$, for $K = 23, 27, 29, 31$, and 33

$R = 6$, for $K = 22, 30, 32$, and 34

The sequences are listed in the Appendix.

To acquire C-correlation, the frame synchronizer correlates the incoming code S_i with C, for all K possible bit shifts, to find the maximum correlation. The performance can be improved by correlating L C-sequences, for each bit shift, and adding their absolute values.

(2) Step II: Frame Synchronization

Once C-synchronization is obtained, S_i is demodulated by C to yield the binary sequence

$$\cdots d_{im_i}, d_{i1}, \cdots, d_{im_i}, d_{i1}, \cdots, d_{im_i}, d_{i1}, \cdots$$

which is composed of the codeword d_i, repeated for each frame. By decoding this bit stream d_i can be found, yielding frame synchronization as well as frame length and data format. As mentioned before, D is a self-synchronizable code; for example, a prefix code (all words start with the same prefix). The d_i-bit signal-to-noise ratio (SNR) is K times the S_i-bit SNR, and thus is of the same order of magnitude as the data-bit SNR. Also, the word length (typically 50–100) is comparable to the size (M) of the dictionary. Therefore, the redundancy of the code is very large, and both prefix and suffix can be chosen to correct several bit errors and assure an almost error-free synchronization.

IV. Serial Frame Synchronization on the Quadrature Channel

In serial frame synchronization a portion of each frame (e.g., the first K bits of the frame) consists of a code sequence, repeated at each frame. It is well known (Ref. 2) that a false synchronization can occur, even without noise, due to a replica of the code sequence, generated by the data. We will present a method which eliminates this possibility, by transferring the serial code to the quadrature channel. That is, during the data period all the energy is transmitted on the in-phase channel, while during the sync period of the frame, all the energy is transferred to the quadrature channel.

Let $S(t)$ be a binary time-multiplexed signal, partitioned into frames of duration T, where each frame starts with an identical sync code $\hat{S}_s(t)$, $t \in [0, T_s]$. Thus,

$$S(t) = \begin{cases} S_s(t), & t \in [0, T_s] + nT \\ S_d(t), & \text{elsewhere} \end{cases} \tag{6}$$

where

$$S_s(t) = \hat{S}_s(t) + \hat{S}_s(t - T) + \hat{S}_s(t - 2T) + \cdots \tag{7}$$

is a periodic repetition of $\hat{S}_s(t)$ with period T, and $S_d(t)$ is the data signal.

If $C(t)$ is the indicator function of $[0, T_s] + nT$, then

$$S_s(t) = S(t)\,C(t)$$

and

$$S_d(t) = S(t)(1 - C(t)) \triangleq S(t)\,\bar{C}(t) \tag{8}$$

Consider the phase modulated signal

$$y(t) = \sin[\omega t + \theta(t)] \tag{9}$$

where ω is the carrier frequency,

$$\theta(t) = \theta S(t) b(t) \overline{C}(t) + \frac{\pi}{2}(1 + S(t)) C(t) \tag{10}$$

$$= \begin{cases} \theta S_d(t) b(t), & \text{during the data signal} \\ \frac{\pi}{2}(1 + S_s(t)), & \text{during the sync signal} \end{cases}$$

and $b(t)$ is a squarewave subcarrier.

Clearly, the carrier is four-phase modulated, where the phases are $\pm\theta$ during the data period and 0 or π during the sync period.

The received signal is

$$r(t) = \sqrt{2P}\, y(t) + n(t) \tag{11}$$

where P is the total received power and $n(t)$ is a zero mean white gaussian noise of one-sided spectral density N_0. From Eqs. (9) to (11),

$$r(t) = n(t) + \begin{cases} S_d(t) b(t) \sin\theta \sqrt{2P} \cos \omega_t + \cos\theta \sqrt{2P} \sin \omega t, & \text{during the data signal} \\ -S_s(t) \sqrt{2P} \sin \omega t, & \text{during the sync signal} \end{cases} \tag{12}$$

Thus, the data signal $S_d(t)$ and the sync code $S_s(t)$ are not only disjoint in time, but transmitted on different channel, that is, $S_d(t)$ is received on the in-phase channel ($\cos \omega t$) while $S_s(t)$ is received on the quadrature channel ($\sin \omega t$).

Assuming perfect coherent demodulation, the output of the inphase and quadrature correlations, $r_1(t)$ and $r_2(t)$, respectively, are (Fig. 3):

$$r_1(t) = n_1(t) + \begin{cases} \sqrt{P} \sin\theta\, S_d(t) b(t), & \text{during data signal} \\ 0, & \text{during sync signal} \end{cases}$$

$$r_2(t) = n_2(t) + \begin{cases} \sqrt{P} \cos\theta, & \text{during data signal} \\ -\sqrt{P} \sin\theta\, S_s(t), & \text{during sync signal} \end{cases}$$

where n_1 and n_2 are independent noise processes.

The modulation index θ determines the amount of unmodulated carrier, transmitted for phase reference.

Since synchronization is acquired and maintained using $r_2(t)$, no false sync occurs due to the random data. Also, since $\sin^2\theta >> \cos^2\theta$ (only a small fraction of the power is transmitted for carrier reference), the SNR in the quadrature channel is large during the sync period of the frame and small during the data period. Therefore, a coarse search of $S_s(t)$ can be made, using a SNR detector, to speed up the acquisition.

Note that the synchronization code does not have to be at the beginning of the frame, but can be distributed anywhere (not necessarily continuously). In this case, $C(t)$ is the indicator function of the synchronization bits, and the same analysis holds.

A. Carrier Tracking

The incoming signal is correlated with the output of the voltage-controlled oscillator (VCO), $\sqrt{2}\cos(\omega t + \phi)$, where ϕ is the phase error in tracking, to yield

$$r_1(t) = n_1(t) + \begin{cases} -\sqrt{P}\cos\theta\sin\phi + \sqrt{P}\sin\theta\cos\phi\, S_d(t)\, b(t), & \text{during data signal} \\ \sqrt{P}\sin\theta\, S_s(t)\sin\phi, & \text{during sync signal} \end{cases} \tag{13}$$

Since $S_s(t)$ can be chosen to have a negligible low frequency component [otherwise, we can transmit $S_s(t)\,b(t)$ instead of $S_s(t)$], the output of the loop filter is the error signal

$$e(t) = n_1(t) + \begin{cases} -\sqrt{P}\cos\theta\sin\phi, & \text{during data signal} \\ 0, & \text{during sync signal} \end{cases} \tag{14}$$

Thus, the VCO is not controlled during the relatively short sync period, which is usually less than 5% of the time. However, the duration T_s of the sync is much shorter than the time constant of the loop, and there is little danger of losing lock during the sync period. The effective loop SNR is

$$\frac{P\cos^2\theta}{N_0 B_L}\left(1 - \frac{T_s}{T}\right) \tag{15}$$

If a Costas loop is used, with bandpass filters for $S_s(t)$ and $s(t)\,b(t)$, then tracking is performed during the whole period.

V. Conclusion

New methods of parallel and serial synchronization for time-multiplexed signals have been presented. In the parallel case, synchronization can be obtained for any of M synchronization codes, which is particularly useful when frames of different lengths have to be transmitted. In the serial case, the false sync due to a replica of the code generated by the random data is eliminated.

Appendix

Sequences With Small Partial Correlations

The following is a list of binary sequences of length K, which have small partial correlations R (Eq. 5) and R (Eq. 4). The sequences are described by the following notation:

$$n_1n_2n_3n_4 \cdots \equiv \underbrace{1\cdots1}_{n_1}\underbrace{0\cdots0}_{n_2}\underbrace{1\cdots1}_{n_3}\underbrace{0\cdots0}_{n_4}\cdots$$

K	Sequence	$\hat{R}$	R
3	2 1	1	1
5	3 1 1	1	1
7	3 2 1 1	1	1
11	3 3 1 2 1 1	1	1
13	5 2 2 1 1 1 1	1	1
14	5 2 2 2 1 1 1	2	2
15	5 2 2 1 1 1 2 1	2	3
	6 2 2 1 1 1 2	2	3
16	3 1 3 4 1 1 2 1	2	4
	5 2 2 2 1 1 1 2	2	4
17	2 2 5 1 1 1 1 2 1 1	2	3
	4 2 2 1 2 1 1 1 1 2	2	3
18	5 1 1 2 1 1 3 2 2	2	2
19	4 3 3 1 3 1 2 1 1	2	3
20	5 1 3 3 1 1 2 1 1 2	2	4
21	6 1 1 3 1 1 2 3 2 1	2	3
	5 1 1 3 1 1 2 3 2 2	2	3
	3 5 1 3 1 2 1 1 1 2 1	2	3
22	8 3 2 1 2 2 1 1 1 1	3	6
23	1 2 2 2 2 1 1 1 5 4 1 1	3	5
24	8 3 2 1 1 1 1 2 2 1 2	3	4
25	3 2 3 6 1 1 1 1 1 2 1 2 1	2	3
26	8 2 1 1 2 2 1 1 1 1 2 3 1	3	4
27	2 1 2 1 1 2 1 3 1 3 1 3 4 2	3	5
28	2 1 2 1 1 2 1 3 1 3 1 3 4 3	2	4
	3 2 3 6 1 1 1 1 1 2 1 2 1 2 1	2	4
29	2 1 2 1 1 2 1 3 1 3 1 3 4 4	3	5
30	7 1 2 2 1 1 1 1 2 1 1 3 4 2 1	3	6
31	3 2 2 3 6 1 1 1 1 1 1 2 1 2 1 3	3	5
32	6 1 3 2 1 3 3 1 1 2 1 2 1 1 3 1	3	6
33	6 3 1 2 3 2 1 1 3 2 1 1 2 2 1 1 1	3	5
	6 3 1 2 2 2 1 1 1 3 2 4 1 1 1 2	3	5
	6 3 1 2 1 1 1 2 1 2 2 3 1 1 1 3 1 1	3	5
34	7 4 2 1 1 2 1 1 2 2 2 2 1 1 1 1 1 1 1	3	6

References

1. Butman, S., and Timor, U., "Interplex—An Efficient Multi-Channel PSK/PM Telemetry System," *Proc. of ICC*, June 1971.

2. Stiffler, J. J., *Theory of Synchronous Communication.* Prentice Hall, Inc., New Jersey, 1971.

3. Turyn, R., "Sequence With Small Correlation," in *Error Correcting Codes.* Edited by Mann. John Wiley & Sons, Inc., New York, 1968.

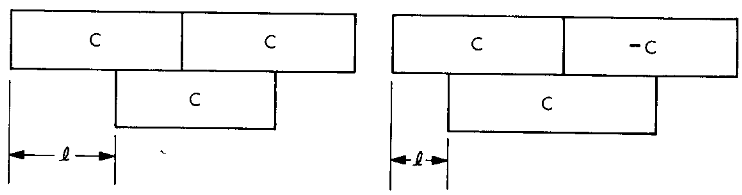

Fig. 1. ℓ**-shift correlation of C with (C, C) and (C, −C)**

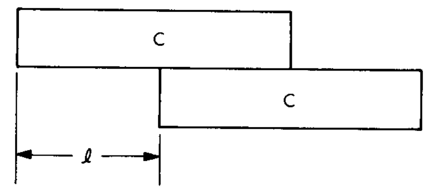

Fig. 2. ℓ**-shift partial correlation of C**

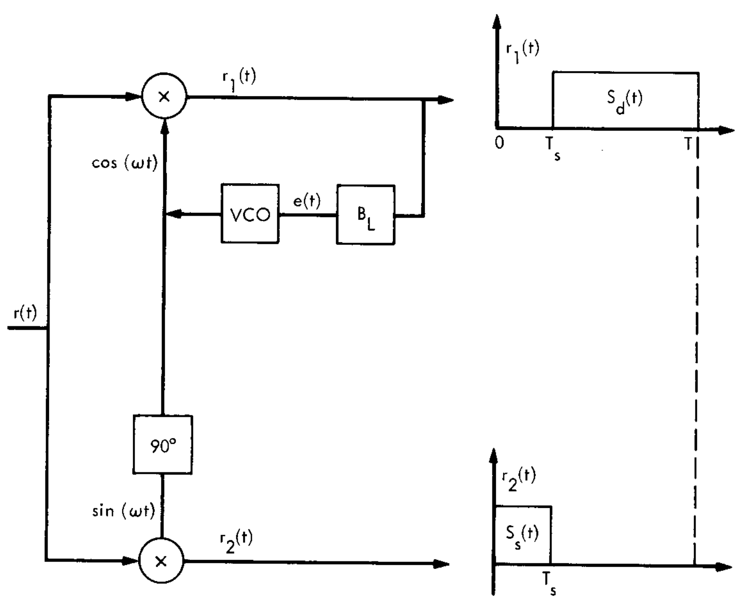

Fig. 3. Receiver for quadrature serial synchronizer